

# PYTHON ENVIRONMENT

---

[http://www.tutorialspoint.com/python/python\\_environment.htm](http://www.tutorialspoint.com/python/python_environment.htm)

Copyright © tutorialspoint.com

Before we start writing our Python programs, let's understand how to setup our Python environment. Python is available on a wide variety of platforms:

- Unix (Solaris, Linux, FreeBSD, AIX, HP/UX, SunOS, IRIX etc.)
- Win 9x/NT/2000
- Macintosh (PPC, 68K)
- OS/2
- DOS (multiple versions)
- PalmOS
- Nokia mobile phones
- Windows CE
- Acorn/RISC OS
- BeOS
- Amiga
- VMS/OpenVMS
- QNX
- VxWorks
- Psion
- Python has also been ported to the Java and .NET virtual machines

## Getting Python:

The most up-to-date and current source code, binaries, documentation, news, etc. is available at the official website of Python:

**Python Official Website :** <http://www.python.org/>

You can download Python documentation from the following site. The documentation is available in HTML, PDF, and PostScript formats.

**Python Documentation Website :** [www.python.org/doc/](http://www.python.org/doc/)

## Install Python:

Python distribution is available for a wide variety of platforms. You need to download only the binary code applicable for your platform and install Python.

If the binary code for your platform is not available, you need a C compiler to compile the source code manually. Compiling the source code offers more flexibility in terms of choice of features that you require in your installation.

Here is a quick overview of installing Python on various platforms:

## Unix & Linux Installation:

Here are the simple steps to install Python on Unix/Linux machine.

- Open a Web browser and go to <http://www.python.org/download/>
- Follow the link to download zipped source code available for Unix/Linux.
- Download and extract files.
- Editing the *Modules/Setup* file if you want to customize some options.
- **run** `./configure` script
- **make**
- **make install**

This will install python in a standard location `/usr/local/bin` and its libraries are installed in `/usr/local/lib/pythonXX` where XX is the version of Python that you are using.

## Windows Installation:

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <http://www.python.org/download/>
- Follow the link for the Windows installer *python-XYZ.msi* file where XYZ is the version you are going to install.
- To use this installers *python-XYZ.msi*, the Windows system must support Microsoft Installer 2.0. Just save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file by double-clicking it in Windows Explorer. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you're ready to roll!

## Macintosh Installation:

MacPython is a Python version available for the Macintosh. Jack Jansen maintains it, and you can have full access to the entire documentation at his Web site.

**Jack Jansen Website :** <http://www.cwi.nl/~jack/macpython.html>

Just go to this link and you will find complete installation detail for Mac OS installation.

## Setting up PATH:

Programs and other executable files can live in many directories, so operating systems provide a search path that lists the directories that the OS searches for executables.

The path is stored in an environment variable, which is a named string maintained by the operating system. These variables contain information available to the command shell and other programs.

The **path** variable is named PATH in Unix or Path in Windows (Unix is case-sensitive; Windows is not).

In Mac OS, the installer handles the path details. To invoke the Python interpreter from any particular directory, you

must add the Python directory to your path.

## Setting path at Unix/Linux:

To add the Python directory to the path for a particular session in Unix:

- **In the csh shell:** type  
setenv PATH "\$PATH:/usr/local/bin/python" and press Enter.
- **In the bash shell (Linux):** type  
export PATH="\$PATH:/usr/local/bin/python" and press Enter.
- **In the sh or ksh shell:** type  
PATH="\$PATH:/usr/local/bin/python" and press Enter.

**Note:** /usr/local/bin/python is the path of the Python directory

## Setting path at Windows:

To add the Python directory to the path for a particular session in Windows:

- **At a command prompt :** type  
path %path%;C:\Python and press Enter.

**Note:** C:\Python is the path of the Python directory

## Python Environment Variables:

Here are important environment variables which can be recognized by Python:

Variable	Description
PYTHONPATH	Has a role similar to PATH. This variable tells the Python interpreter where to locate the module files you import into a program. PYTHONPATH should include the Python source library directory and the directories containing your Python source code. PYTHONPATH is sometimes preset by the Python installer.
PYTHONSTARTUP	Contains the path of an initialization file containing Python source code that is executed every time you start the interpreter (similar to the Unix .profile or .login file). This file, often named .pythonrc.py in Unix, usually contains commands that load utilities or modify PYTHONPATH.
PYTHONCASEOK	Used in Windows to instruct Python to find the first case-insensitive match in an import statement. Set this variable to any value to activate it.
PYTHONHOME	An alternative module search path. It's usually embedded in the PYTHONSTARTUP or PYTHONPATH directories to make switching module libraries easy.

## Running Python:

There are three different ways to start Python:

### (1) Interactive Interpreter:

You can enter **python** and start coding right away in the interactive interpreter by starting it from the command line. You can do this from Unix, DOS, or any other system which provides you a command-line interpreter or shell window.

```
$python          # Unix/Linux
or
python%         # Unix/Linux
or
C:>python       # Windows/DOS
```

Here is the list of all the available command line options:

Option	Description
-d	provide debug output
-O	generate optimized bytecode (resulting in .pyo files)
-S	do not run import site to look for Python paths on startup
-v	verbose output (detailed trace on import statements)
-X	disable class-based built-in exceptions (just use strings); obsolete starting with version 1.6
-c cmd	run Python script sent in as cmd string
file	run Python script from given file

## (2) Script from the Command-line:

A Python script can be executed at command line by invoking the interpreter on your application, as in the following:

```
$python script.py          # Unix/Linux
or
python% script.py         # Unix/Linux
or
C:>python script.py       # Windows/DOS
```

**Note:** Be sure the file permission mode allows execution.

## (3) Integrated Development Environment

You can run Python from a graphical user interface (GUI) environment as well. All you need is a GUI application on your system that supports Python.

- **Unix:** IDLE is the very first Unix IDE for Python.
- **Windows:** PythonWin is the first Windows interface for Python and is an IDE with a GUI.
- **Macintosh:** The Macintosh version of Python is called MacPython and also available from the main website, downloadable as either MacBinary or BinHex'd files. MacPython includes an IDE.

Before proceeding to next chapter, make sure your environment is properly setup and working perfectly fine. If you are not able to setup the environment properly then you can take help from your system admin.

All the examples given in subsequent chapters have been executed with Python 2.4.3 version available on CentOS flavor of Linux.